WHAT IS CLAIMED IS:

Sub A1

1. A method for preventing unauthorized access to hardware management information comprising:

receiving a request for hardware component information in a service processor disposed in a hardware component as an open session request from a requesting client application, which request passed to the service processor external to an operating system controlling the hardware component;

transmitting from the service processor a challenge string to the requesting client application;

receiving in the service processor a challenge response from the requesting client application;

comparing the challenge response to an expected response to the challenge string; and

transmitting hardware component information to the requesting client application.

2. The method according to claim 1, wherein the challenge string includes a session identification number unique to each session.

3. The method according to claim 1, wherein the challenge response includes a session identification number unique to each session and assigned by the service processor.

4. The method according to claim 1, wherein the challenge response includes a sequence number that increments with every new message.

5. The method according to claim 1, wherein the challenge response includes a hash number, wherein the hash number is a function of one or more of the following: the challenge string, the session identification number, the sequence number and a password.

a1

6. The method according to claim 1, further comprising examining each packet received from the client application for one or more of the following: the session identification number, the sequence number and a hash number.

7. The method according to claim 6, wherein the hash number is a function of one or more of the following: the session identification number, the sequence number and the packet itself.

8. A method for preventing unauthorized access to hardware management information comprising:

transmitting a request for hardware component information to a service processor disposed in a hardware component as an open session request from a requesting client application;

passing the request to the service processor external to an operating system controlling the hardware component;

receiving from the service processor a challenge string at the requesting client application;

transmitting to the service processor a challenge response from the requesting client application; and

receiving from the service processor an authentication response to the requesting client application based on a comparison of the challenge response from the requesting client application and an expected challenge response calculated in the service processor.

9. The method according to claim 8, wherein the challenge string includes a session identification number assigned by the service processor, which session identification number is unique to each session, and the challenge response includes the session identification number.

10. The method according to claim 9, wherein the challenge response includes a sequence number that increments with every new message from the requesting client application.

11. The method according to claim 8, wherein the challenge response includes a hash number calculated by the requesting client application, and the hash number is a function of one or more of the following: the challenge string, the session identification number, the sequence number and a password.

12. The method according to claim 8, further comprising transmitting with each packet sent by the client application one or more of the following: the session identification number, the sequence number and a hash number, and the hash number is a function of one or more of the following: the session identification number, the sequence number and the packet itself.

13. An apparatus for authenticating a client application requesting access to a particular component among a plurality of components, comprising:

a remote access port; and

a service processor disposed in the particular component, coupled to the remote access port, and in response to a remote request for information about the particular component received as an open session request through the remote access port external to a host operating system, the service processor is programmed to:

transmit a challenge string to a requesting client application;

compare a challenge response received from the requesting client application with an expected response to the challenge; and

transmit an authentication response to the requesting client application based on the comparison.

14. The apparatus according to claim 13, wherein service processor assigns a session identification number unique to each session and transmits the session identification number to the requesting client application in the challenge string.

15. The apparatus according to claim 14, wherein the service processor reviews the challenge response to determine if it contains the session identification number transmitted in the challenge string.

16. The apparatus according to claim 13, wherein the service processor compares a sequence number included in the challenge response against previously received sequence numbers and ignores the challenge response if it does not include a sequence number in correct sequence.

17. The apparatus according to claim 13, wherein the service processor compares a hash number received in the challenge response with an expected hash calculated by the service processor and transmits a success or failure message depending upon a result of the comparison.

18. The apparatus according to claim 17, wherein the hash includes one or more of the following: the challenge string, the session identification number, the sequence number and a password.

19. The apparatus according to claim 13, wherein the service processor examines each packet sent by the client application for one or more of the following: the session identification number, the sequence number and a hash number, wherein the hash number is a function of one or more of the following: the session identification number, the sequence number and the packet itself.

20. A system for accessing hardware component information from a computer, comprising:

a service processor disposed in the computer;

a server remotely coupled to the service processor in the computer;

a client application to execute on the server, wherein the service processor authenticates requests from the client application requesting access to the service processor's host hardware module, which request bypasses an operating system of the computer, and the service processor in response to a request for access to the host hardware module is programmed to:

transmit a challenge string to a requesting client application;

compare a challenge response received from the requesting client application with an expected response to the challenge; and

transmit an authentication response to the requesting client application based on the comparison.

21. The system according to claim 20, wherein each of the service processors assigns a session identification number unique to each session and transmits the session identification number to the requesting client application in the challenge string.

22. The system according to claim 20, wherein each of the service processors reviews the challenge response to determine if it contains the session identification number transmitted in the challenge string.

23. The system according to claim 20, wherein each of the service processors compares a sequence number included in the challenge response against previously received sequence numbers and ignores the challenge response if it does not include a sequence number in correct sequence.

24. The system according to claim 20, wherein each of the service processors compares a hash number received in the challenge response with an expected hash calculated by the service processor and transmits a success or failure message depending upon a result of the comparison.

25. The system according to claim 24, wherein the hash includes one or more of the following: the challenge string, the session identification number, the sequence number and a password.

26. The system according to claim 20, wherein each of the service processors examines each packet sent by the client application for one or more of the following: the session identification number, the sequence number and a hash number, wherein the hash number is a function of one or more of the following: the session identification number, the sequence number and the packet.

27. A method for verifying integrity of a data packet comprising:

receiving the data packet in a service processor disposed in a hardware component from a client application, which data packet passes external to an operating system and a system processor otherwise controlling operation of the hardware component;

receiving with the data packet a keyed hash of the data packet; and

comparing the keyed hash with the data packet to an expected keyed hash.

28. The method according to claim 27, wherein the keyed hash is a function of one or more of the following: a session identification number, a sequence number, a password and the data packet.

29. A method for verifying integrity of a data packet comprising:

transmitting a data packet to a service processor disposed in a hardware component from a client application, which data packet passes external to an operating system and system processor otherwise controlling the hardware component;

calculating a keyed hash of the data packet; and

transmitting to the service processor the keyed hash along with the data packet.

30. The method according to claim 29, wherein the keyed hash is a function of one or more of the following: a session identification number, a sequence number, a password and the packet.

~~31. An apparatus for preventing unauthorized access to hardware management information comprising a computer readable media having programming instructions encoded thereon, instructing a processor to:~~

~~receive a request for hardware component information in a service processor disposed in a hardware component as an open session request, which request passes external to an operating system controlling the hardware component;~~

~~transmit from the service processor a challenge string to the requesting client application;~~

~~receive in the service processor a challenge response from the requesting client application;~~

~~compare the challenge response to an expected response to the challenge; and~~

~~transmit from the service processor an authentication response to the requesting client application based on the comparison.~~

~~32. An apparatus for preventing unauthorized access to hardware management information comprising a computer readable media having programming instruction encoded thereon instructing a processor to:~~

~~transmit a request for hardware component information to a service processor disposed in a hardware component as an open session request from a requesting client application, which request passes external to an operating system controlling the hardware component;~~

~~receive from the service processor a challenge string at the requesting client application;~~

~~transmit to the service processor a challenge response from the requesting client application; and~~

~~receive from the service processor an authentication response to the requesting client application based on a comparison of the challenge response from the requesting client application and an expected challenge response calculated in the service processor.~~

a

~~33. An apparatus for verifying integrity of a data packet comprising a computer readable media having programming instructions encoded thereon instructing a processor to:~~

~~receive the data packet and a keyed hash in a service processor disposed in a hardware component from a client application, which data packet and keyed hash pass external to an operating system and a system processor otherwise controlling operation of the hardware component;~~

~~calculate an expected a keyed hash of the data packet; and~~

~~compare the received keyed hash with the expected keyed hash.~~